①

IMENTATION PAGE

Form Approved
OMB No. 0704-0188

ı is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, ıing and reviewing the collection of information Send comments regarding this burden estimate or any other aspect of this ıing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson ıd to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 30 December 1993	3. REPORT TYPE AND DATES COVERED **Technical Report,** 7/1/93 - 12/31/93

4. TITLE AND SUBTITLE

ADAPTIVE DECISION MAKING AND COORDINATION
IN VARIABLE STRUCTURE ORGANIZATIONS

5. FUNDING NUMBERS

N00014-93-1-0912

6. AUTHOR(S)

Alexander H. Levis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Center of Excellence in Command, Control, Communications and Intelligence
George Mason University
Fairfax, Virginia 22030

8. PERFORMING ORGANIZATION REPORT NUMBER

GMU/C3I-147-IR

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Arlington, Virginia 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Progress in research on coordination in distributed decision making organizations with variable structure is reported. The problem of consistency and completeness of the set of decision rules used by an organization is addressed by modeling the rule base by a Colored Petri Net and then analyzing the static and dynamic behavior of the net. The design problem is addressed by focusing on algorithms that relate the structural properties of the Petri Net model of the organization to its behavioral characteristics.

14. SUBJECT TERMS

Decision making, Organization Theory, Coordination, Colored Petri Nets, Rule based systems

15. NUMBER OF PAGES

22

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT **Unclassified**	18. SECURITY CLASSIFICATION OF THIS PAGE **Unclassified**	19. SECURITY CLASSIFICATION OF ABSTRACT **Unclassified**	20. LIMITATION OF ABSTRACT **UL**

CENTER OF EXCELLENCE IN
COMMAND, CONTROL, COMMUNICATIONS AND INTELLIGENCE

GEORGE MASON UNIVERSITY
Fairfax, Virginia 22030

SEMI-ANNUAL TECHNICAL REPORT

for the period

1 July 1993 - 31 December 1993

for

ADAPTIVE DECISION MAKING AND COORDINATION IN VARIABLE STRUCTURE ORGANIZATIONS

Grant Number N00014-93-1-0912

Submitted to:
Dr. W. S. Vaughan, Jr. (3 copies)
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217-5000

Submitted by:

Alexander H. Levis
Principal Investigator

Copies to:
Director, Naval Research Laboratory
Administrative Grants Office, ONR
Defense Technical Information Center

December, 1993

Report #: GMU/C3I-147-IR

1. PROGRAM OBJECTIVES

The objective of this research is the investigation of several issues related to coordination in organizations. In particular, an organization is coordinated through direct and indirect means. The direct means includes the set of decision rules that the organization members use and the commands that they issue to each other. Indirect means include the dissemination of information within the organization; for example, organization members may share information or they may inform each other as to the actions they plan to take or decisions they have made. Coordination becomes a complex issue in variable structure organizations. Not only do the decision rules and the information architecture have to work for each fixed structure, but the designer has to deal with the problem, a metaproblem, of coordinating the variability. This becomes a particularly difficult problem in organizations that exhibit substantial complexity and redundancy in their information structure. The redundancy is necessary both for robustness and for flexibility and reconfigurability. In order to address these problems two main tasks were defined; they are described in the next section. In addition, some basic work in algorithms and Colored Petri Nets needs to be done to develop tools and techniques for supporting the analysis and design.

2. STATEMENT OF WORK

The statement of work, as described in the proposal, is given below.

Task 1: Consistency and completeness in distributed decision making

Develop a methodology for analyzing and correcting the set of decision rules used by an organization with distributed decision making. The methodology is to be based on the modeling of the distributed decision rules in the form of a Colored Petri Net and on the analysis of the net using s-invariant properties and occurrence graphs. The ability to test and correct the set of rules has direct impact on the extent of coordination needed in an actual organization and the resulting communication load.

Task 2: Variable Structures: Heuristic rules in the Lattice algorithm constraints

Develop a methodology for selection of the degrees of redundancy and complexity and a procedure for checking the validity of the different degrees (to be derived from the DFS algorithm of Andreadakis) and incorporate them in the Lattice Algorithm. Generalize the approach to multilevel organizational structures and to variable structures, where variable structures are obtained by folding together different fixed structures. The real focus of the task is to introduce redundancy and complexity as a way of containing the dimensionality problems inherent in flexibility and reducing the coordination requirements.

Design a Symbolic Interface for the Lattice algorithm. The interface would have the capability of interpreting natural language inputs entered by the user and will include some symbolic processing. The system will generate the interconnection matrices used as input to the Lattice algorithm. The designer would then use the various tests described in this proposal (such as the DFS algorithm) to check the validity of the interconnection constraints and to make required modifications.

Task 3: Information Dissemination

Semiannual progress reports will be submitted in accordance with ONR requirements. The results of this research will appear in thesis reports and in technical papers to be presented at professional meetings and published in archival journals. In addition, oral presentations will be given periodically as arranged with ONR.

3. RESEARCH PLAN

The research plan describes the strategy for meeting the program objectives. Specifically the research plan is organized around a series of specific well-defined research tasks that are appropriate for theses at the master's and Ph.D. level. Individual students are assigned to each task under the supervision of the principal investigator. Additional staff from the C3I Center are included in the project whenever there is specific need for their expertise..

The empirical development of the set of decision rules used by the members of an organization (the DMs) to perform a wide variety of tasks, or the decomposition of normative or prescriptive rules so that they can be assigned to different DMs, can result in knowledge bases that are inconsistent, incomplete and partially erroneous due to a number of human and implementation related factors involved in the process. A task that addresses a fundamental question in integrating the system and coordination models, namely, the consistency and completeness of the embedded rules, was initiated. The focus of the task is the development of a methodology for analyzing and correcting the set of decision rules used by an organization with distributed decision making. The methodology is based on the modeling of the distributed decision rules in the form of a Colored Petri Net and on the analysis of the net using s-invariant properties and Occurrence graphs. This work is being done by A. Zaidi as part of his Ph.D. thesis.

In preparation for task 2, several subordinate tasks have been initiated that focus on the development of algorithms and tools. One problem that has come up a number of times and needed clarification is the discrepancies between the theoretical constructs of Petri Nets, especially concurrency, and the software implementation of Petri Net analysis and design tools. In order to understand the behavioral properties of the Petri Net software, a short project was carried out in which the early work of Grevet and Levis (1988) on coordination was re-examines in the context of Colored Petri Nets. This project was completed during the summer by Ms. Hedy Rashba as a master's project.

A second issue that affects directly the use of the lattice algorithm in task 2 is the computation of s-invariants and the extension of the algorithm to the determination of deadlocks and traps in the organizational structure. Ms. Jin is completing her Master's thesis on this subject.

With the second of these two projects is completed, work will start on the specifics of Task 2.

4. STATUS REPORT

This section presents a discussion of the work carried out during the indicated period of performance. It is organized along the task structure described in Sections 2 and 3.

4.1 CONSISTENCY AND COMPLETENESS IN DISTRIBUTED DECISION MAKING

Following are some of the issues that have been the focus of research effort during the reporting period.

- Techniques to reduce the size of the problem
- Occurrence graph analysis

A detailed research plan for this task is shown in Figure 1. The nodes in the chart describe major issues or sub-issues, which are either investigated or are required to be researched. The plan is likely to change with time as new results are obtained.

A simple Colored Petri Net (CPN) representation was developed which can be used to transform decision rules represented in terms of expressions in First Order Predicate Calculus (FOPC). To automate the process of transforming decision rules to their equivalent CPN representation, a software program was written during the reporting period of this document. The algorithm for transforming decision rules to CPN representation is implemented on Design CPN™ using ML™. The program takes a text file with decision rules as the input and transforms the rules to a Colored Petri Net. In addition, a procedure, if invoked, works on the CPN and identifies the incomplete cases in the rule base. The program has been tested and it appears to work. Further tests will be carried out as the work progresses.

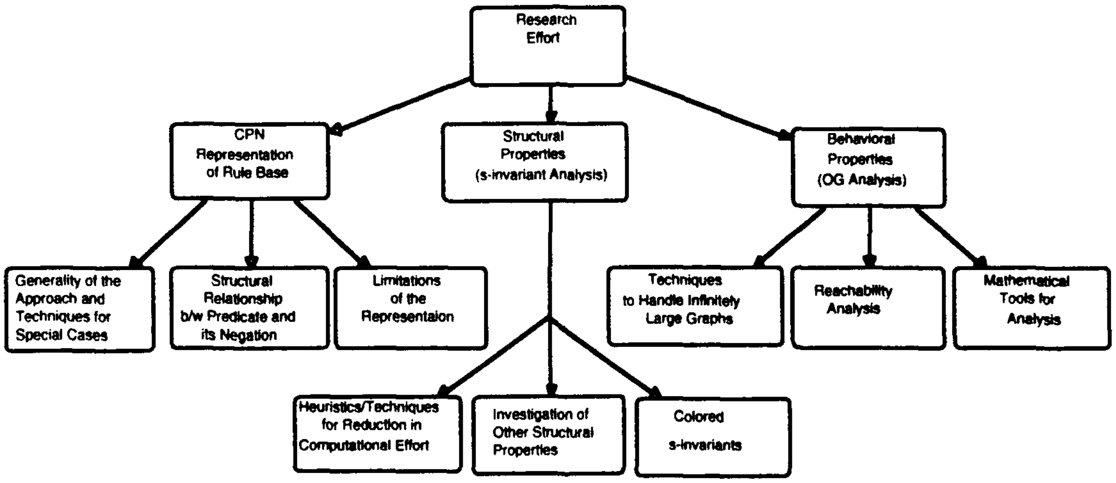

Research Effort
CPN Representation of Rule Base
Structural Properties (s-invariant Analysis)
Behavioral Properties (OG Analysis)
Generality of the Approach and Techniques for Special Cases
Structural Relationship b/w Predicate and its Negation
Limitations of the Representaion
Techniques to Handle Infinitely Large Graphs
Reachability Analysis
Mathematical Tools for Analysis
Heuristics/Techniques for Reduction in Computational Effort
Investigation of Other Structural Properties
Colored s-invariants

Figure 1 Possible Directions of Research

The analysis of the underlying Ordinary Petri Nets (PN) of the CPN using s-invariant properties has been found to reveal a number of problems in the set of decision rules being modeled by these nets. The results of the s-invariant analysis have been reported in earlier report (TDM initiative). During this reporting period, a program has been implemented on Design CPN™ that constructs an incidence matrix of the drawn CPN. The incidence matrix can then be input to the s-invariant algorithm already implemented in C.

4.1.1 Problem Reduction Techniques

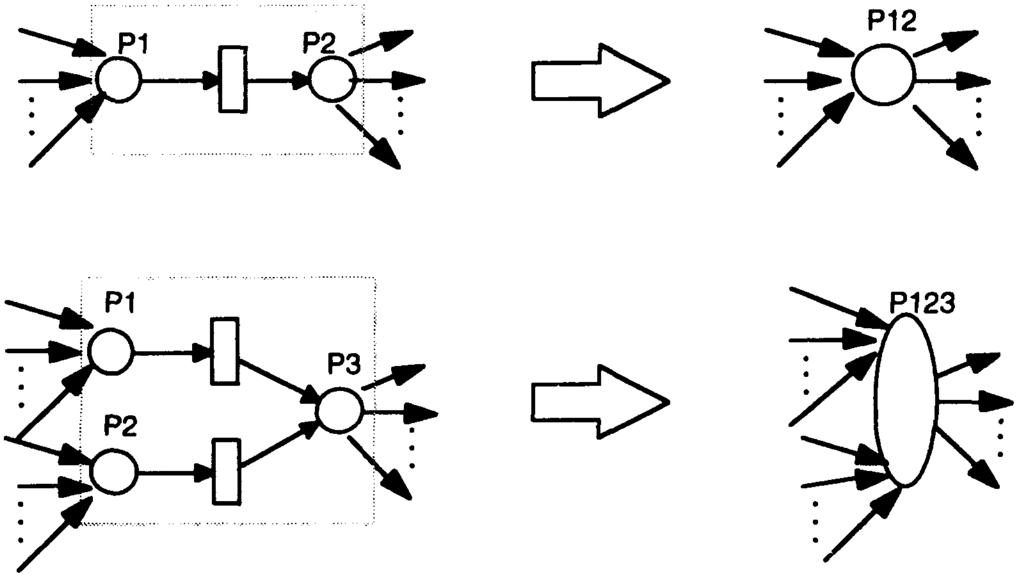
A large set of decision rules, when transformed to an equivalent CPN representation, yields a large net. The computational effort required to calculate the s-invariants makes the approach infeasible for such large sized nets. The research effort was directed on an approach to break down the initial problem into several *non-interacting* sub-problems, and then calculating the invariants for these smaller problems individually. The *FindPath* algorithm [V. Y. Jin, 1989] has been found promising for achieving this goal. The algorithm, if applied to an output, gives all the paths starting from all the inputs and terminating at the selected output, thus isolating the sub-net corresponding to a particular output from the rest of the net. The s-invariant algorithm can now be applied to this small net. An iterative application of the FindPath and s-invariant algorithms will yield all the invariants of the original net. The approach is promising since it not only breaks the computationally large problem into smaller problems but also yields a very few number of redundant results. During this reporting period, a new version of the FindPath algorithm was implemented, which works on the graph itself and redraws the relevant portions (relative to a given output) of the net on a separate page.

In a parallel effort, a number of PN reduction techniques have been investigated to reduce the size of the net. Several such techniques have already been reported in the literature on PN [Jensen, 1991], [Andreadakis, 1990]. The focus of the effort was to apply the transformation techniques used in First Order Predicate Calculus (FOPC) to an equivalent PN representation. The result of this effort reduces the PN to a smaller net.

In FOPC, if we have a conditional expression of the form, p1 —> p2, where pi $\in$ rule base, then predicates p1 and p2 are semantically identical if and only if p1 -/-> pi, where i $\neq$ 2 and pj -/-> p2, where j $\neq$ 1.

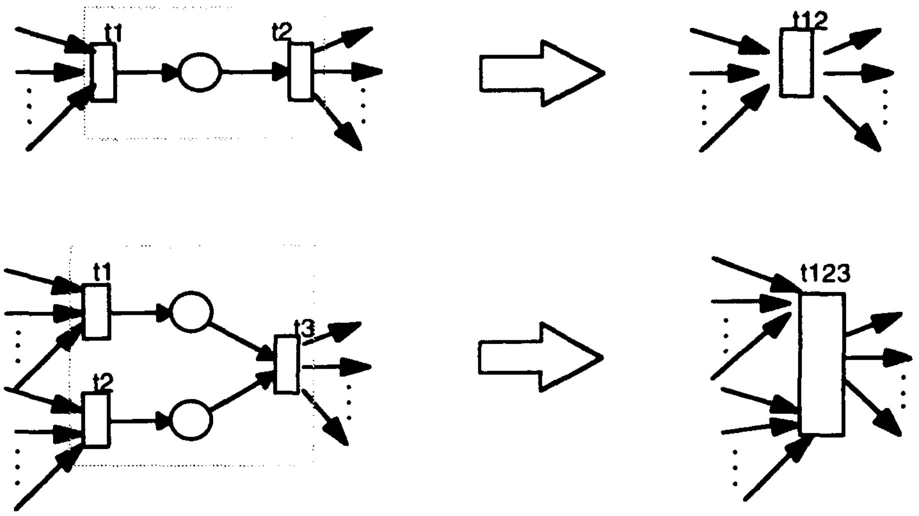
The above mentioned result from FOPC yields the two PN reduction techniques shown in Figure 2a. In the figure, the two places labeled as p1 and p2 are compounded into a single *compound* place labeled as p12. In a more general case, an entire segment of a net can be replaced by a single compound place as long as all the boundary nodes of the sub-net are all places and have either input arcs coming from outside the boundary or output arcs going out of the boundary of the sub-net.

Similarly, Figure 2b shows the dual techniques for reducing the net by compounding sub-nets into compound transitions.



(a)

Figure 2 Petri Net Reduction Techniques



(b) Duals

Figure 2 Petri Net Reduction Techniques

The Petri Net reduction techniques presented in Figure 2 can also be viewed as producing an abstract description of the rule base. The approach gives a hierarchical methodology to solve the problem. The hierarchical problem solving approach, however, requires more thought, which at this moment is beyond the scope of the research. A number of other transformations in FOPC can also be used to reduce the net, but such techniques may lead to time consuming and more involved search procedures for PN structures. However, the two techniques, if applied iteratively, would reduce the net substantially and may cover cases that are complex and difficult to find at first inspection.

4.2. Occurrence Graph Analysis

An Occurrence graph associated with a PN and an initial marking represents all possible markings of the net (recall that the marking of the net represents the state of the system) that can be achieved from the initial one. It has been reported in an earlier report that the generated Occurrence graphs can reveal a number of the remaining (according to the list of problems listed in the proposal) problems in the set of decision rules. This report presents some of the specific analyses that help find some of the inconsistent and redundant cases. The calculation of Occurrence graphs can yield very large graphs making it difficult to look for the combinations of markings (states) that correspond to erroneous decision rules. The two PN reduction techniques mentioned in the previous section help resolve this issue to a certain extent. However, a number of other factors such as presence of loops in a small net can have a substantial effect on the size of the Occurrence graph. One possible approach is to isolate the circular rules that give rise to these loops before starting to construct an Occurrence graph. The circular rules can be found during the s-invariant analysis. It seems at this moment that this issue is application-dependent and requires more effort to resolve. Another issue raised in the previous report was the determination of initial markings. With no knowledge of the set of valid inputs, the search for the valid input vectors in the input space is a combinatorial problem; the calculation of the Occurrence graphs for all such candidate input vectors is computationally infeasible. In order to resolve the issue of initial marking, an approach is being investigated that has, so far, been found promising and appropriate. The approach is to reverse the directions of the arcs in the net and then initializing the net by putting tokens in each output place (the set of output is assumed to be known). After initializing the net, an Occurrence graph is generated and checked for states corresponding to the erroneous cases. Some of the new results obtained during the reporting period are outlined below.

Algorithm for Redundancies and Subsumed Rules

1. Construct an ordinary PN of the original CPN by replacing arc expressions by constants and redefining all color sets in the Global Declaration node.
2. Construct a set, A, consisting of places with multiple input arcs (starting from the set of valid outputs).
3. From A select a place, p0, apply the FindPath algorithm to the net with the selected place as the output. The result of the algorithm is a net called PN'.
4. Remove from A all the places that are present in PN'.
5. Reverse the direction of the arrows of PN'.
6. Initialize PN' with a token in p0 and draw an abstract Occurrence graph using the OG Analyzer^{TM}.
7. Query the constructed Occurrence graph for nodes with multiple input arcs.
8. Find the firing vectors corresponding to each input arc/path, which bring the system from the initial state to the one under consideration. Remove duplicate firing vectors.
9. The remaining firing vectors identify the set of redundant rules.
   Example 1 illustrates these steps.
10. Query the constructed Occurrence graph in step 6 for nodes that are subsets/supersets of each other.
11. Find the firing vectors corresponding to each node, which bring the system from initial state to the ones under consideration. Remove duplicate firing vectors.
12. The remaining firing vectors identify the set of subsumed rules.
    Example 2 illustrates these steps.
13. Repeat the steps as long as there are elements in A.

Example 1 (Redundant Rules)

Figure 3 shows the output of the FindPath algorithm applied to a hypothetical rule base. The algorithm described above starts with step 4. Figure 4 shows the PN' initialized and all arrows reversed in direction. Figure 5 shows the Occurrence graph constructed for the PN' in Figure 4. The following two distinct firing vectors were found corresponding to three input arcs to the bottom node in Figure 5.

$$F1 = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}\begin{array}{l}t1\\t2\\t3\\t4\end{array} \qquad F2 = \begin{bmatrix}0\\1\\1\\1\end{bmatrix}\begin{array}{l}t1\\t2\\t3\\t4\end{array}$$

The corresponding redundant rules are:

t1: p1, p2, p3, p4 —> p0

and

t2: p1, p2 —> p5
t3: p3, p4 —> p6
t4: p5, p6 —> p0

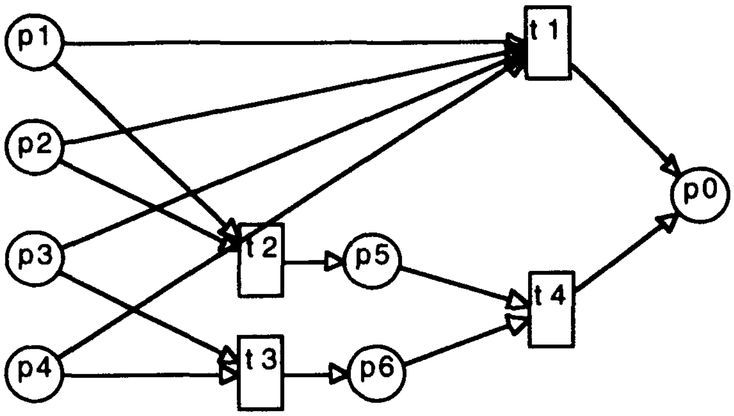


Figure 3 PN'

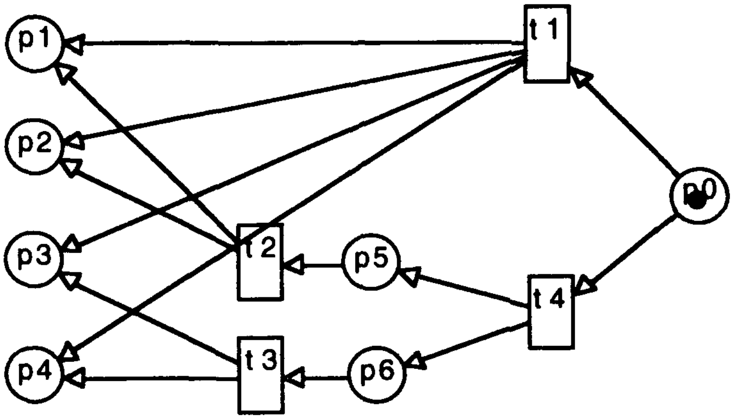


Figure 4 PN' Initialized and Arrows Reversed

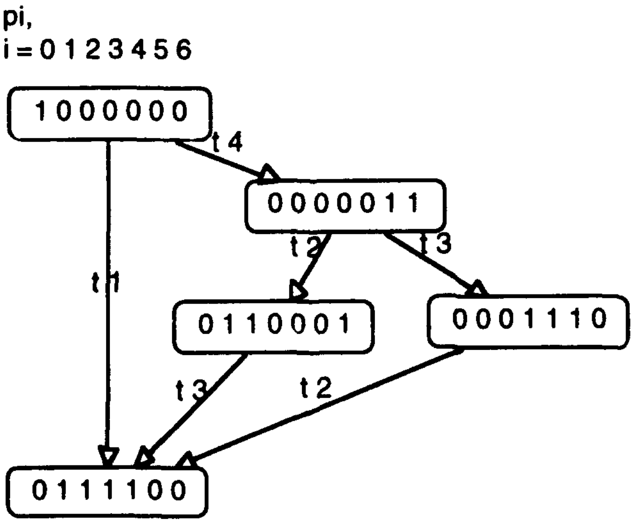


Figure 5 Occurrence graph for PN' in Figure 4

Example 2 (Subsumed Rules)

Figure 6 shows the output of the FindPath algorithm applied to a hypothetical rule base. The example illustrates steps 10 and 11 of the algorithm. Figure 7 shows the PN' initialized and all arrows reversed in direction. Figure 8 shows the Occurrence graph constructed for the PN' in Figure 7. The shaded node s1 is a subset of the shaded node s2. The following two distinct firing vectors were found to correspond to these two states.

$$F1 = \left[\begin{array}{c}1\\0\\0\\0\end{array}\right]\begin{array}{l}t1\\t2\\t3\\t4\end{array} \qquad F2 = \left[\begin{array}{c}0\\1\\1\\1\end{array}\right]\begin{array}{l}t1\\t2\\t3\\t4\end{array}$$

The corresponding rules are:

t1: p1, p2, p3 —> p0

subsumes

t2: p1, p2 —> p5
t3: p3, p4 —> p6
t4: p5, p6 —> p0

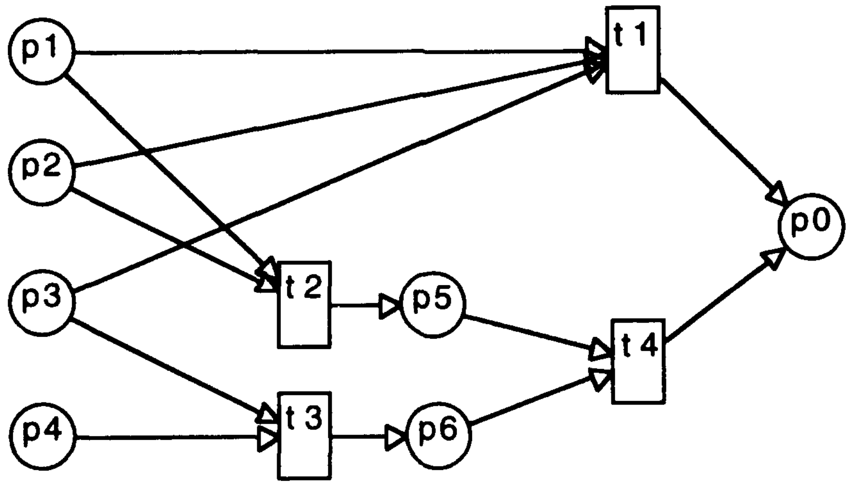


Figure 6 PN'

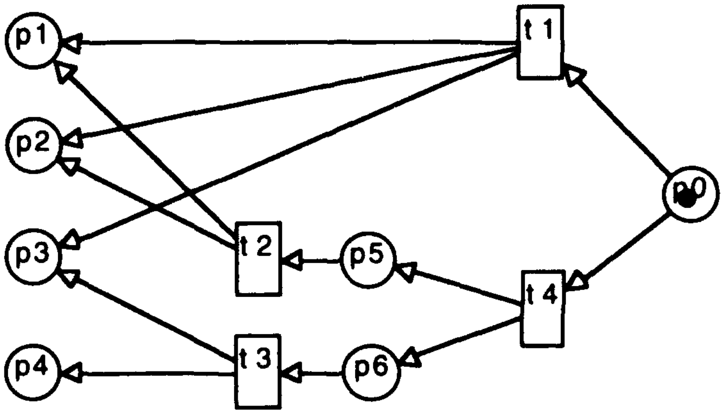


Figure 7 Modified PN' with Initial Marking

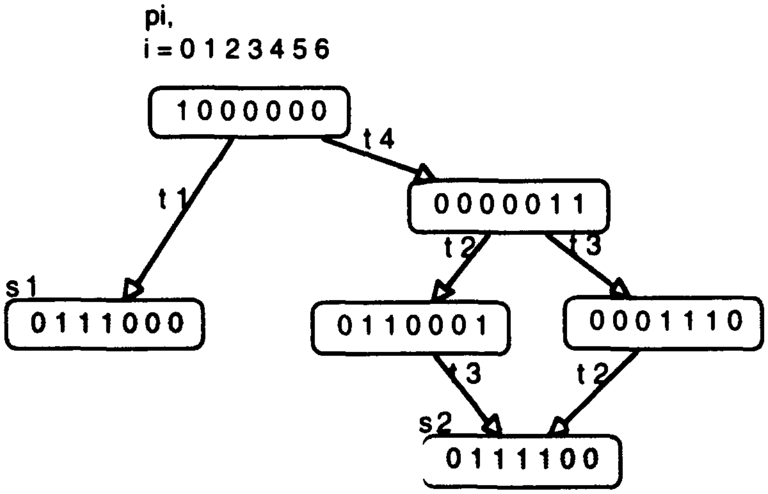


Figure 8 Occurrence Graph for PN' in Figure 7

Algorithm for Conflicting Rules

1. Construct an ordinary PN of the original CPN by replacing arc expressions by constants and redefining all color sets in the Global Declaration node.
2. Construct a set, B, consisting of places which have their negations present in the net (starting from the set of valid outputs).
3. From B select a place, p0 and its negation ~p0, merge these two places into a single place P0. Apply the FindPath algorithm to the net with the P0 as the output. The result of the algorithm is called PN'.
5. Reverse the direction of the arrows of PN'.
6. Initialize PN' with a token in P0 and draw an abstract Occurrence graph using OG Analyzer™.
7. Follow steps 7 - 11 from the algorithm for redundant and subsumed cases. If any of such cases is found, the corresponding firing vector identifies the presence of conflicting rules. Example 3 illustrates the steps.
8. Repeat the steps as long as there are elements in B.

Example 3 (Conflicting Rules)

Figure 9 shows the output of the FindPath algorithm applied after the place p0 and its negation ~p0 are merged together into a single place P0, the resulting place P0 is shown with dotted line. The modified net is initialized with a token in P0 and the directions of all arrows are reversed. An Occurrence graph is constructed for the net, the graph would resemble the graph in Figure 5. The two distinct firing vectors (same as in Example 1) would give the following conflicting rules.

t1: p1, p2, p3, p4 —> ~p0

and

t2: p1, p2 —> p5
t3: p3, p4 —> p6
t4: p5, p6 —> p0

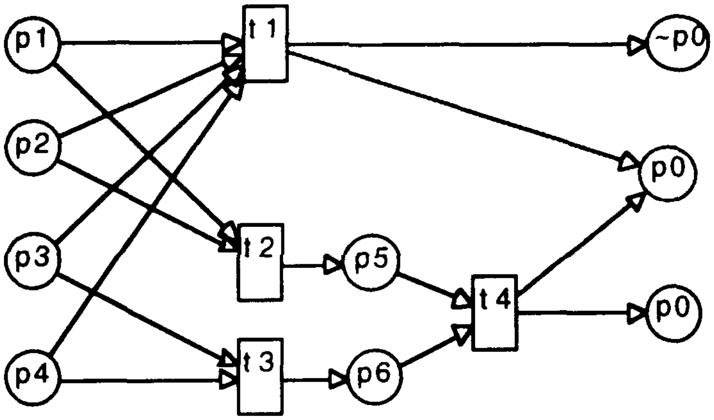


Figure 9 PN' with Merged Place p0

4.2 VARIABLE STRUCTURES: HEURISTIC RULES IN THE LATTICE ALGORITHM CONSTRAINTS

4.2.1 Problems of Concurrency and Coordination in Decision Making Organizations

This task utilized Colored Petri Nets (CPN) as supported by Design/CPN™ to revisit aspects of an earlier analysis of a two-person hierarchical organization. The project focused on how Colored Petri Nets both (1) uncovered a need to address the various stages of decision making (as modeled by transitions in the Petri Net terminology) when faced with large amounts of information requiring simultaneous evaluation (concurrent or having *multiple enablements* in Petri Net terminology), and (2) provided a means to represent these various stages of information. Colored Petri Nets, it is shown, can capture the example with a level of flexibility that none of the earlier methodologies could. This study also revealed that the underlying assumptions of Petri Net software can highlight different types of information processing.

The results of the study have been documented in a technical report (#8.1 in the documentation list) that is being submitted separately.

4.2.2 Deadlocks and Traps

Petri Net theory has been a successful tool for the study of concurrency in discrete event systems because it allows their detailed and precise mathematical representation. Analysis of Petri Nets can reveal important information about the structural and dynamic behavior of the modeled system, and this information can then be used for evaluation and to suggest improvements or changes. One of the analysis techniques is the determination of deadlocks and traps. *Deadlocks* are sets of places which remain empty once they have lost all tokens. Conversely, *traps* are sets of places which remain marked once they have gained at least one token. Deadlock and trap theory has a direct application to the study of coordination in decision making organizations. In general, the presence of deadlocks and/or traps can indicate pathological conditions in the organization's design.

Research on deadlock and trap problems started in the early 70's, as indicated in the references (Silva, 1990). Basic deadlock and trap problems in Petri Nets have been resolved to a certain extent, but computation of deadlocks and traps has been limited to strongly connected free choice nets, which are a subset of ordinary Petri Nets. No work that has appeared in the literature has extended the deadlock/trap theory to hierarchical Petri Nets.

Because of the complexity of possible net transformations and the limited types of deadlocks and traps that can be calculated by the currently available algorithms, a more general and easy to implement algorithm is needed to find deadlocks and traps in Ordinary Petri Nets. Also, as real systems are often designed in a hierarchical modular structure, an algorithm is necessary to compute deadlocks and traps in hierarchical Petri Nets.

The basic structural and behavioral features of Petri Nets were introduced in C.A. Petri's papers in early 1960's. The notions of deadlock and trap were first defined by Hack in 1972 (Esparza, 1990). The requirement that all deadlocks remain marked can be structurally achieved if all deadlocks contain initially marked traps; this is known as Commoner's property (Best, 1987). Commoner's property has been proved to be necessary and sufficient for the liveness of Free Choice and Extended Free Choice systems. It is also sufficient for Asymmetric Choice systems (Best, 1987).

The practical application of the theory of deadlocks and traps requires to have efficient algorithms for their computation. The classical methods use Boolean equations (Silva, 1990), sometimes translated into linear inequalities (Esparza, 1990). An alternative approach was studied by Lautenbach (1987), in which deadlocks and traps were related to special P-semiflows of an associated net, thus opening up the possibility of applying the S-invariant method to the calculation of strongly connected deadlocks and traps in strongly connected Free Choice nets. This algorithm requires transformation of the net into an associated marked graph, which may become a very tedious procedure for very large Petri Nets.

This research task addresses two major problems :

1. Development of algorithms that can find deadlocks and traps in ordinary Petri Nets with no structural restrictions.

3. Extension of the deadlock and trap algorithms to hierarchical Petri Nets. In order to do this, matrix relations between hierarchical Petri Nets and the equivalent non-hierarchical Petri Nets need to be established.

In the first stage of this research, only Ordinary Petri Nets are considered. Once these problems have been resolved, the hierarchical Colored Petri Net case will be addressed. The goal is to generalize the solution to the deadlock and trap problem; one way to do this is to develop a general algorithm that can find all deadlocks and traps without complex transformation of the original nets, which is the currently available technique. While the current methods are restricted to strongly connected Free Choice nets, this research addresses the general, unrestricted case. Also, this work will extend deadlock and trap theory to hierarchical Petri Nets.

Ordinary Petri Nets

A Petri Net is a bipartite directed graph represented by a quadruple N= (P, T, I, O) where

- P = {p1, ..., pn} is a finite set of *places.*,
- T = {t1, ..., tm} is a finite set of *transitions.*
- I is a mapping P × T -> {0,1} corresponding to the set of directed arcs from places to transitions.
- O is a mapping P × T -> {0,1} corresponding to the set of directed arcs from transitions to places.

An example of a Petri Net is shown in Figure 10. Places are represented by circles and transitions by bars.

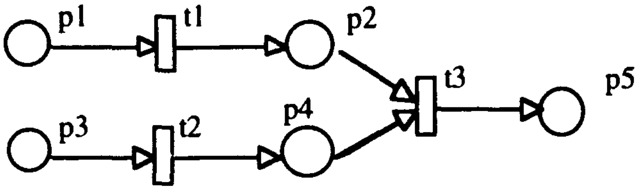


Figure 10. Ordinary Petri Net

A marking of a Petri Net is a mapping : p-> {0, n} which assigns a non-negative integer number of tokens to each place of the net.

A transition is enabled by a given a marking if and only if each of its input places contains at least one token provided each input arc represents a single connection between the place and the transition. When a transition is enabled, it can fire: one token is removed from each input place and one is added to each output place.

Consider the Petri Net in Fig. 11 with the indicated marking.

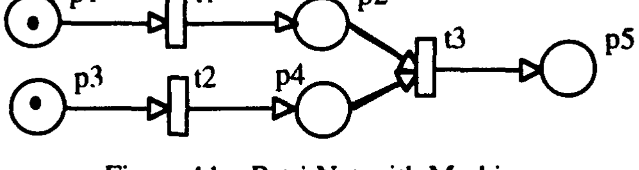


Figure 11. Petri Net with Marking

In Fig. 11, if t1 and t2 fires, then the resulting marking is shown in Figure 12.

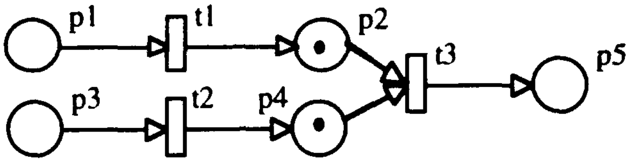


Figure 12 Petri Net after Firing

Transitions t3 is enabled. If t3 fires, the new marking is shown in Figure 13.

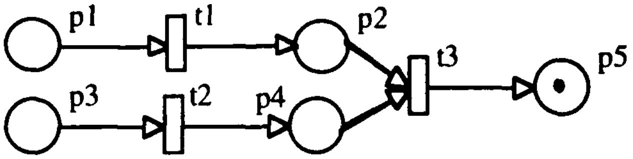


Figure 13. Petri Net after Second Firing

Hierarchical Petri Nets allow the designer to create a large model composed of many submodels, and isolate a segment to study its details without disturbing or altering the entire structure. They also provide a modular approach towards modeling a complex system. This feature is vital for designing complex organizations. Basically, there are two ways to construct hierarchical Petri Nets. One is by a compound transition, and the other is by a compound place. As compound transition are more often used in the construction of hierarchical Petri Nets, only compound transitions will be used in this research.

Compound Transition

If a subnet of a Petri Net model is replaced by a single transition, the single transition is termed *compound transition*. It represents the aggregated effect of the processes represented by the transitions of the subnet. The model with compound transitions describes the system at a higher degree of abstraction than the one without them.

Figure 14 shows a Petri Net model of a system in which the system's functionality is described at the most detailed level. The dotted box contains the processes that are to be aggregated. In Figure 15 the outlined subnet is shown replaced by a single transition - a compound transition denoted by the label "HS". The subnet that represents the compound transition at a *subpage* is shown in Figure 16. The term subpage is used in *Design /CPN*™, a commercially available software package for Hierarchical Petri Nets, to denote pages which contain the subnets replaced by compound transitions and compound places.

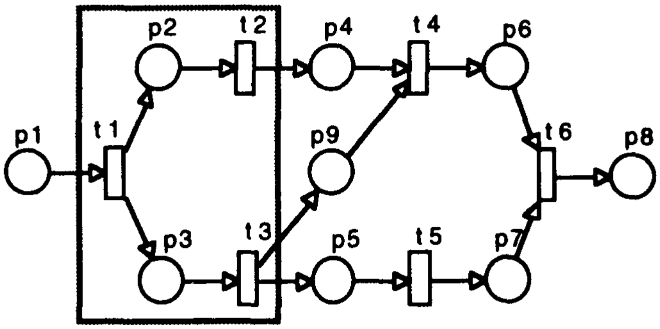


Figure 14 Detailed Description of a System

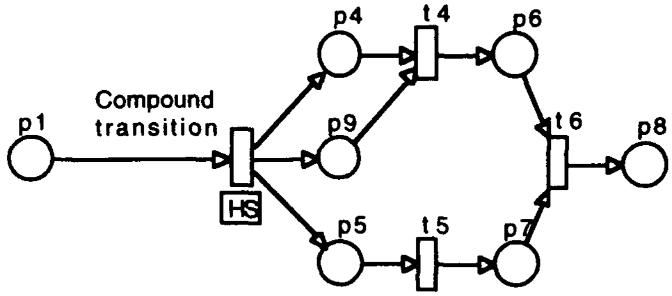


Figure 15 Systen:'s Description with a Compound Transition

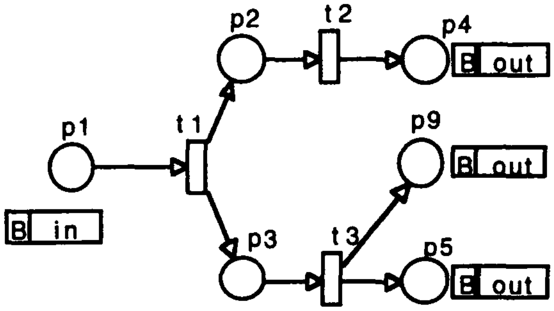


Figure 16 Subpage Representation of the Compound Transition

The places, in Figure 16, with label "B in" or "B out" represent the *port nodes*. Port nodes are defined to be the input and output places of the subnet; its connections with the uncompound net. On the other hand, all those places whose input and output transitions are defined within the subnet are not port nodes. Port nodes are used to preserve the connectivity of the original net. They model the sockets for the places that exist in the preset and postset of the compound transition in the system's net. The places p1, p4, p5, and p9 in Figure 15 are defined as port nodes in Figure 16.

Deadlocks And Traps Of Petri Nets

A Petri Net representing an organization consists of two parts, a net and a marking. The net models the static structure of the organization, and the marking represents its state. By saying "structure" of a Petri Net, we mean that the net has properties that are dependent on the way places and transitions of the net are interconnected by the flow relations, and these properties are independent of the marking of the net.

The dynamic behavior of a net depends on the initial marking of the net. A marking of the net is a distribution of tokens over the places. In Ordinary Petri Nets, tokens are indistinguishable from each other. The dynamics of the net are captured through the firing of transitions. It is obvious that the behavior of marked net is closely related to the structure of the net. The behavior properties include, for example, deadlocks and traps, the existence of invariants, boundedness, and many others. Deadlocks and traps are related to the liveness of a Petri Net.

When investigating the behavior characteristics of a Petri Net, it is important to consider parts of the net which will never be marked or which will never lose all their tokens. Such parts will be considered and, in particular, we shall consider those in which such situations are easily recognizable. A set D of places will never be marked again, after losing all tokens, if and only if no transition which contains in its postset a place belonging to D may ever fire again. In particular, this is the case when all these transitions also contain a place belonging to D in their preset. A set of places that meets this condition is called a deadlock. Deadlocks are critical for liveness analysis, because transitions may never be enabled again if they contain places of an unmarked deadlock in their preset.

There are also net parts which will never lose all tokens again after they have once been marked. This is the dual to the deadlocks; it is the case in which for some set of places, Q, when every transition removing tokens from Q also puts at least one token onto Q. If Q fulfills this condition, then Q is called a trap.

The preset of a node is the set of all nodes having arcs leading to it. The postset of a node is the set of all nodes that it has arcs leading to. The formal definition of preset and postset are given as follows:

Definition 1 *Preset and Postset*

Let N= (P, T, I, O) be a Petri Net.

Let $x \in P \cup T$. The preset $^*x$ and postset x^* are given by

$^*x = \{ y \in P \cup T \mid (y, x) \in (P \times T) \cup (T \times P)\};$

$x^* = \{ y \in P \cup T \mid (x, y) \in (P \times T) \cup (T \times P)\};$

The preset of a set of nodes is the union of the presets of these nodes.
The postset of a set of nodes is the union of the postset of these nodes.

Definition 2 *Deadlocks and traps*

Let N= (P, T, I, O) be a Petri Net.

1. A non empty set $D \subseteq P$ is called a deadlock iff $^*D \subseteq D^*$.
2. A non empty set $Q \subseteq P$ is called a trap iff $Q^* \subseteq {}^*Q$.

Deadlock and trap properties :

Let N= (P, T, I, O) with an initial marking Mo,

1. Non-marked deadlocks remain non-marked.
2. Marked traps remain marked.
3. The union of deadlocks is still a deadlock; the union of traps is still a trap.

Deadlocks, Traps And Petri Nets Hierarchies

There are four classes of ordinary Petri Nets that form a Petri Net hierarchy (Rozenberg and Thiagarajan, 1986) as shown in Figure 17.

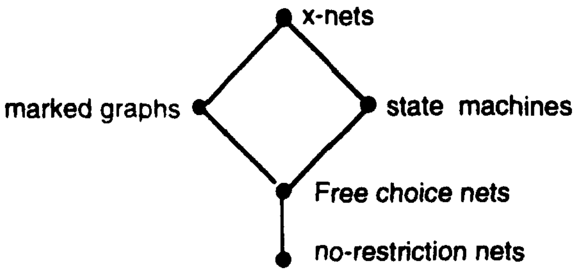


Figure 17 Petri net hierarchy

- Deadlocks and Traps in X-nets

  X-nets are a class of nets such that each place has at most one input and one output transition, and each transition has at most one input place and one output place.(Rozenberg, and Thiagarajan, 1986) Deadlocks and traps are easy to find in x-nets. In general, for strongly connected x-nets, deadlocks and traps are the cycles in an x-net. As the strongly-connected x-net itself is a simple cycle, so the deadlock and trap for the x-net are actually the same -- the x-net itself.

- Deadlocks and Traps for Marked Graphs

  For strongly connected marked graphs, deadlocks and traps are the cycles in the nets .

- Deadlocks and Traps for State Machines

  For state machines, deadlocks and traps are the same -- the net itself.

- Deadlocks and Traps for Free Choice (FC) nets

  To find out deadlocks and traps for Free Choice nets, we need certain algorithms. Lautenbach (1987) proposed an algorithm that can calculate strongly connected deadlocks and traps for strongly connected FC nets by applying the S-invariant algorithm. This algorithm requires that the net be transformed into an Associated Marked graph. Details of this algorithm can be found in Lautenbach (1987).

- Deadlocks and traps in other nets

  Current deadlock/trap algorithms can be handled up to strongly connected Free Choice nets. The deadlocks and traps calculated are strongly connected components.

APPROACH

The approach is based on the mathematical definition of deadlocks and traps, which is dependent on the place preset and place postset relations in a Petri Net. Then, by means of matrix representation, these relations are specified in a Preset-Postset Relation (PPR) matrix. Deadlocks and traps can be found by checking the PPR matrix.

The Preset-Postset Relation Matrix

As the preset and the postset of a place set are both transitions sets (could be empty), so the preset and postset relation of a place set is actually a relation between a place set and a transition sets. Therefore, we can set up a matrix that shows the relations of preset and postset.

The Preset-Postset Relation (PPR) matrix is defined as follows :

Given a Petri net N= (P, T, I, O), Choose a place set $PS_i \subseteq P$, Suppose n_r place sets are chosen, the Preset-Postset relation is represented by a $n_r \times m_r$ matrix denoted as R (PPR matrix). n_r is the number of place sets, m_r is the number of transitions of net N. R_{ij} denotes the element in the i-th row and the j-th column in the matrix, it can has one of the following values:

- $R_{ij} = -1$, if place set PS_i has the property that transition $t_j \in PS_i^*$, and t_j may or may not be in $^*PS_i$.
- $R_{ij} = 1$, if place set PS_i has the property that transition $t_j \in {}^*PS_i$, but $t_j \notin PS_i^*$.
- $R_{ij} = 0$, if place set PS_i has the property that transition tj $\notin PS_i^*$ and $t_j \notin {}^*PS_i$.

Example Given the net shown in Figure 18,

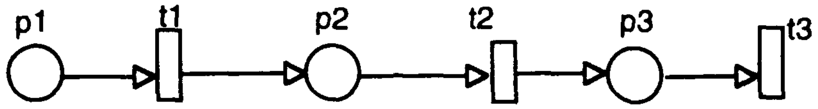


Figure 18. A Petri Net, N1

The Preset-Postset Relation matrix is:

$$\mathbf{R} = \begin{array}{r|ccc|ll} & t1 & t2 & t3 & & \\ \hline \{p1\} & -1 & 0 & 0 & \text{row 1} & {}^*\{p1\}=\{\ \},\ \{p1\}^*=\{t1\} \\ \{p2\} & 1 & -1 & 0 & \text{row 2} & {}^*\{p2\}=\{\ t1\ \},\ \{p2\}^*=\{t2\} \\ \{p3\} & 0 & 1 & -1 & \text{row 3} & {}^*\{p3\}=\{\ t2\ \},\ \{p3\}^*=\{t3\} \\ \{p2, p3\} & 1 & -1 & -1 & \text{row 4} & {}^*\{p2, p3\}=\{\ t1,t2\ \},\ \{p2, p3\}^*=\{t2, t3\} \\ \{p1, p2\} & -1 & -1 & 0 & \text{row 5} & {}^*\{p1, p2\}=\{\ t1\ \},\ \{p1, p2\}^*=\{t1, t2\} \\ \{p1, p2, p3\} & -1 & -1 & -1 & \text{row 6} & {}^*\{p1, p2, p3\}=\{\ t1, t2\},\ \{p1, p2, p3\}^*=\{t1, t2, t3\} \end{array}$$

Figure 19. The Preset-Postset Relation Matrix of net N1

In this PPR matrix, we can see the preset and postset relation of place sets. By checking the matrix R, if a row j has all its elements be either -1 or 0 , then its corresponding place set is a deadlock. In this case, row 5 is such a row. So {p1, p2} is a deadlock, while {p2, p3} is not a deadlock.

From Figure 19 it can be seen that the row 5, which represents the place set {p1, p2} preset and postset relations can be derived from row 1 and row 2. Row 4 can be derived from row 2 and row 3. This derivation is illustrated as follows :

Place set	PPR element (t1 column)	Preset and Postset relations
{p1}	-1	t1 ∈ $\{p1\}^*$, t1 may or may not be ∈ $^*\{p1\}$
{p2}	1	t1 ∈ $^*\{p2\}$, t1 $^*\{p1\}$ ∉ $\{p2\}^*$
{p1, p2}	(-1)+ 1 = (-1)	t1 ∈ $\{p1, p2\}^*$, t1 may or may not be ∈ $^*\{p1, p2\}$ in this case, t1 ∈ $^*\{p1, p2\}$.

From the illustration, it can be seen that {p1, p2}'s PPR element in column t1 can be obtained from {p1} and {p2}'s PPR elements in column t1. Rules of operation are therefore set up to derive the additional place set rows in the PPR matrix. If place sets that may related to deadlocks and traps are derived in the PPR matrix, deadlocks and traps can be found. The commutative operation rules are defined on the PPR matrix elements, as follows:

1. 1 + 1 = 1;
2. 0+ 0 =0;
3 1 + 0 = 1;
4 (-1) + (-1) =-1;
5 (-1) + 1 = -1;
6 (-1) + 0 = -1;

The trap problem is the dual of deadlock problem. To calculate the trap supports for the net N, we can apply the deadlock algorithm but replace the incidence matrix C by the transpose of C_d (incidence matrix of the dual net).

The Deadlock/ Trap Algorithm

The deadlock/trap algorithm being developed is based on the preset-postset relation matrix and some special "addition" operation rules of the matrix. The goal of this algorithm is to start from the incidence matrix that describes the structure of a Petri Net, and then try to find out place sets that satisfy the deadlocks definition. To do this, we need to perform certain operations on the original incidence matrix. The deadlock algorithm steps are shown in the following flow chart :

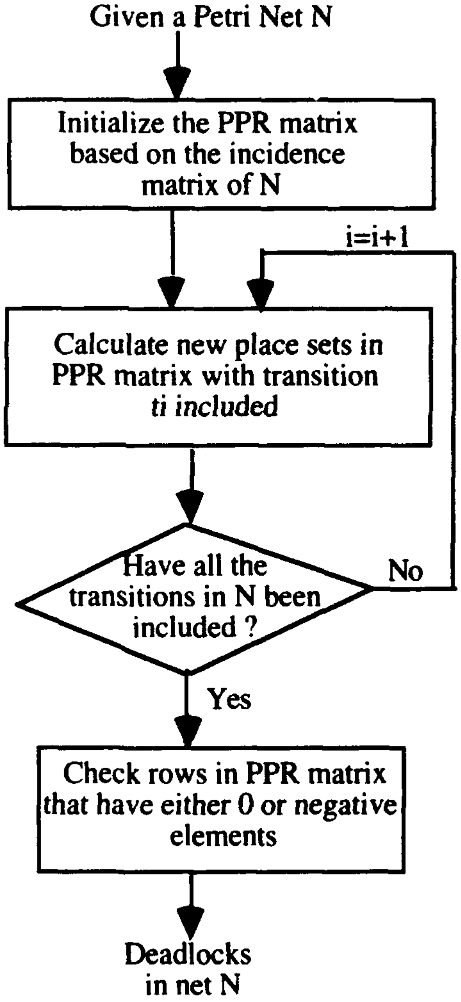

Given a Petri Net N
Initialize the PPR matrix based on the incidence matrix of N
i=i+1
Calculate new place sets in PPR matrix with transition *ti included*
Have all the transitions in N been included ?
No
Yes
Check rows in PPR matrix that have either 0 or negative elements
Deadlocks in net N

Figure 20 Preliminary Deadlock Algorithm Flow Chart

The goal of this research is to find the deadlocks and traps of the unfolded net directly from the deadlocks/traps found in hierarchically structured superpage nets and subpage nets without applying the deadlock/trap algorithm to the large incidence matrix of the unfolded net. The key issue is to find out the incidence matrix relations of superpage nets, subpage nets and the equivalent non-hierarchical Petri Net (unfolded net). The deadlocks relation between subpage nets, superpage nets and unfolded net should be embedded in their incidence matrix relations. These relations are reflected in the port nodes. The preliminary steps of this work are shown in the following flow chart (Fig. 21):

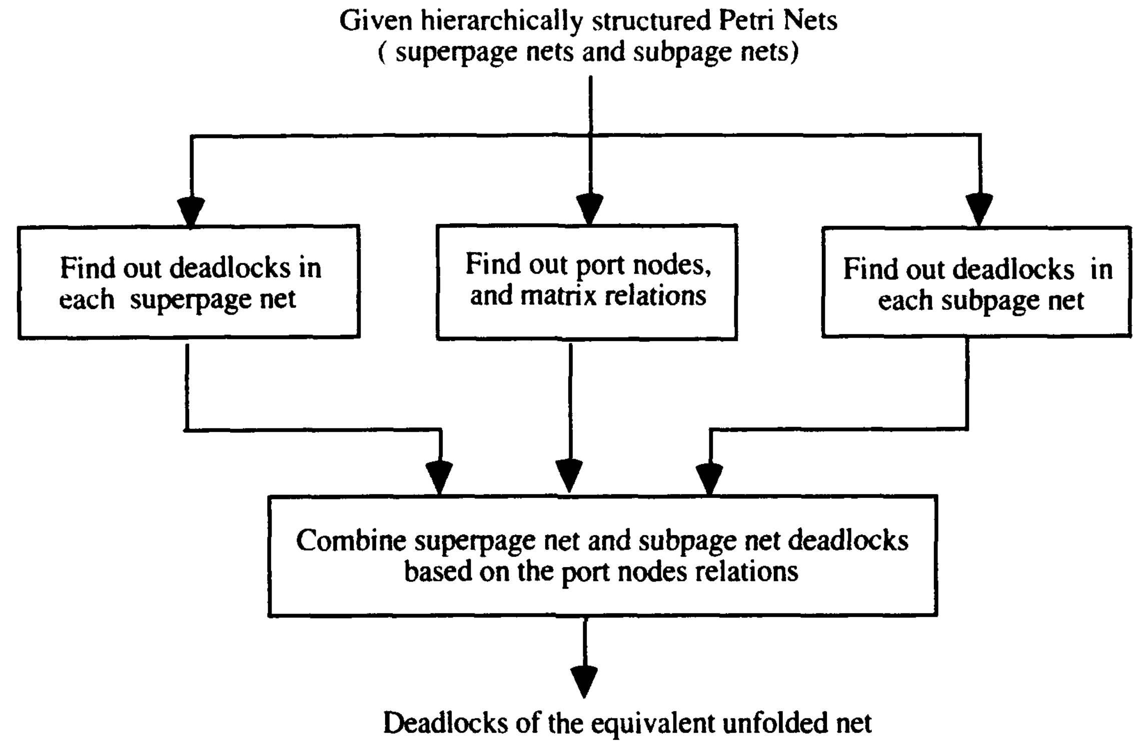

Given hierarchically structured Petri Nets
(superpage nets and subpage nets)
Find out deadlocks in each superpage net
Find out port nodes, and matrix relations
Find out deadlocks in each subpage net
Combine superpage net and subpage net deadlocks based on the port nodes relations
Deadlocks of the equivalent unfolded net

Figure 21. Preliminary Hierarchical Petri Net Deadlock Algorithm Flow Chart

REFERENCES

Best, E. (1987) " The Free Choice Hiatus ", in *Advanced Course on Petri Nets 1987*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany

Beth, E. and Thiagarajan. P.S. (1987) " Some Classes of Live and Safe Petri Nets", in *Concurrency and Nets*, Voss, Genrich, Rozenberg(eds.), Springer-Verlag, Berlin, Germany

Datta, A. and Ghosh, S. (1984) "Synthesis of a Class of Deadlock-Free Petri Nets" , Journal of the ACM. Vol 31, No.3, July 1984

Esparza, J. and Silva, M. (1990) " A Polynomial-time Algorithm to Decide Liveness of Bounded FC Nets" , Hildesheimer Informatik--Berichte, Dec. 1990

Kemper, P. and Bause, F (1992) " An Efficient Polynomial-Time Algorithm to Decide Liveness and Boundedness of Free Choice Nets", Lecture Notes in Computer Science, Vol 616, Springer-Verlag, Berlin, Germany

Lautenbach, K. H. (1987) "Linear Algebraic Calculation of Deadlocks and Traps" in *Concurrency and Nets*, Voss, Genrich, Rozenberg(eds.), Springer-Verlag, Berlin, Germany

Martinez, J. and Silva, M. (1982) " A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri Net", in *Application and Theory of Petri Nets*, Girault, Reisig (eds.), Springer, Informatik-Facheberichte No.52, 1982

Peterson, J. L. (1981) *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Englewood Cliffs, NJ.

Reisig, W. (1985) *Petri Nets, An Introduction*. Springer-Verlag, Berlin, Germany.

Rozenberg, G. and Thiagarajan, P.S. (1986) " Petri Nets: Basic Notions, Structure and Behavior", Lecture Notes in Computer Science vol. 224, P585-668 Springer-Verlag, Berlin, Germany

Singhal, M. (1989) Deadlock Detection in Distributed Systems", IEEE Computer, November 1989, p37-p48, 1989

Thiagarajan, P.S. and Voss, K. (1984) " A Fresh Look at Free Choice Nets", Information and Control vol. 61/2, p85-p113,1984

Zaidi, S. Abbas K. (1991) "On the Generation of Multilevel Distributed Intelligence Systems Using Petri Nets". GMU/C3I-112-TH, M.S. Thesis, C3I Center, George Mason University, Fairfax, VA.

5.0 MEETINGS

There were no meetings during this reporting period.

6.0 CHANGES

There were no changes during this period.

7.0 RESEARCH PERSONNEL

7.1 Research Personnel - Current Reporting Period

The following persons participated in the effort during the reporting period:

Prof. Alexander H. Levis	Principal Investigator
Prof. K. C. Chang*	(August 1993)
Mr. Didier Perdu	Graduate Student (Ph.D.)
Mr. Abbas Zaidi	Grad. Res. Assistant (Ph.D.)

Ms Jenny Jin	Grad. Res. Assistant (MS)
Ms Hedy Rashba*	Graduate Student (MS)
Ms Azar Sadigh*	Grad. Res. Assistant
Ms. Cynthia Johnson*	Graphics Designer

Prof. Chang participated for one man month to assist in questions of algorithm development.

Ms Rashba contributed at no cost to the project; she did her work as part of her Master's project.

Ms. Sadigh was supported for one month in the summer to complete her work on adaptive decision making. A Master's thesis is in preparation with Prof. Lehner as the advisor.

Ms. Johnson, a graphics designer, is working on the development of techniques for presenting the dynamic behavior of the Colored Petri Net models of organizations.

7.2 Research Personnel - Previous Reporting Periods.

None.

7.3 Personnel Changes

None.

8.0 DOCUMENTATION

1. H. R. Rashba "Problems of Concurrency and Coordination in Decision Making Organizations," Report GMU/C3I-143-R, C3I Center, George Mason University, Fairfax, VA, September 1993.